คอมพายเลอร์ ล่ามคอมพิวเตอร์

กิติพงษ์ ฤทธิบุตร

เมื่อพูดถึงคำว่า "คอมพายเลอร์" (compiler) ก็คงจะไม่เป็นคำพูดที่แปลกหูสำหรับท่านที่เป็นโปรแกรมเมอร์หรือผู้ที่ติดตามและศึกษาเกี่ยวกับคอมพิวเตอร์มาบ้าง แต่หลายท่านก็อาจจะยังไม่รู้จักหรือเข้าใจในเรื่องนี้มากนัก ในขณะเดียวกันก็อาจมีอีกหลาย ๆ ท่านที่มีความสนใจใคร่รู้ว่าที่แท้จริง "คอมพายเลอร์" นั้นคืออะไรกัน(แน่ ๆ) และเขาทำมันขึ้นมาได้อย่างไร มีวัตถุประสงค์อะไร แล้ว "คอมพายเลอร์" ที่ว่านี้มีความสำคัญมากน้อยแค่ไหนในการใช้เครื่องคอมพิวเตอร์ แต่ก็มาติดตรงที่ว่าหนังสือหรือตำราที่เกี่ยวกับเรื่องนี้ที่เป็นภาษาไทยนั้นไม่รู้จะไปหาอ่านได้ที่ไหน ก็เลยถือโอกาสนำเรื่องนี้มาคุยกันโดยมีความตั้งใจที่จะช่วยส่งเสริมสนับสนุนให้มีการเริ่มต้นที่จะหันมาศึกษาค้นคว้าในเรื่องนี้กันมากขึ้น อันอาจจะเป็นการนำไปสู่การมี "คอมพายเลอร์" รุ่นใหม่ ๆ ที่เหมาะสมกับสภาพงานและผู้ใช้มากกว่าของเดิมที่มีอยู่ขึ้นมาใช้ในบ้านเรา ซึ่งถ้าความหวังนี้เป็นจริงขึ้นมาก็จะเป็นประโยชน์อย่างยิ่งกับผู้ใช้เครื่องคอมพิวเตอร์ทั้งหลาย หรืออย่างน้อยก็สามารถทำให้ผู้ที่ศึกษาค้นคว้าในเรื่องนี้มีความเข้าใจเกี่ยวกับ "คอมพายเลอร์" ดีขึ้น และอาจนำไปประยุกต์ใช้ในการเขียนโปรแกรมสำเร็จรูปเพื่อการค้า หรือเขียนโปรแกรมสำหรับงานเฉพาะอย่าง (application programs) ได้อย่างดี หรืออย่างน้อยที่สุดก็เป็นไปตามสุภาษิตไทยว่า "รู้ไว้ใช่ว่าใส่ปาแบกหาม"

ก่อนถึง "คอมพายเลอร์"

ในยุคก่อนที่จะมีการผลิตเครื่องคอมพิวเตอร์ขึ้นมาใช้นั้น คนเราได้พยายามหาเครื่องมือที่จะใช้เพื่อช่วยในการคำนวณ ซึ่งเริ่มจากการนับนิ้วมือ สิ่งของแทนนิ้ว และเริ่มมีการใช้เครื่องคำนวณชนิดต่าง ๆ ตั้งแต่ลูกคิด จนในที่สุดถึงเครื่องคอมพิวเตอร์ ซึ่งเครื่องคอมพิวเตอร์อิเล็กทรอนิกส์เครื่องแรกของโลกเริ่มใช้ในปี 2488 (ENIAC:Electronic Numerical Integrator And Calculator) และก็ได้มีการพัฒนาเทคโนโลยีด้านนี้เรื่อยมาจนถึงในยุคปัจจุบัน ซึ่งมีเครื่องคอมพิวเตอร์ให้เลือกใช้อย่างมากมาย และในยุคแรก ๆ ของการนำเครื่องคอมพิวเตอร์มาใช้งานนั้น การสั่งงานเครื่องทำได้ด้วยความยากลำบาก เพราะการสั่งให้เครื่องทำงานหรือประมวลผลนั้นจะต้องเขียนคำสั่งด้วยภาษาเครื่อง (machine language) หรือภาษาแอสเซมบลี และโดยที่การประมวลผลหรือการคำนวณสำหรับงานชิ้นหนึ่ง ๆ นั้น อาจจะประกอบด้วยคำสั่งต่าง ๆ มากมาย ทั้งจำนวนและชนิดของคำสั่ง เช่น คำสั่งในการนำข้อมูลเข้า/ออก คำสั่งให้ทำการคำนวณในลักษณะต่าง ๆ คำสั่งการตัดสินใจตามเงื่อนไข คำสั่งในการเคลื่อนย้ายข้อมูล และคำสั่งติดต่อกับอุปกรณ์รอบข้าง เช่น ดิสค์ไดรฟ์และเครื่องพิมพ์ เป็นต้น ซึ่งการที่จะต้องเขียนโปรแกรมสั่งงานให้เครื่องทำการประมวลผลตามต้องการ ด้วยภาษาเครื่องและแอมเซมบลีนั้นเป็นเรื่องยุ่งยาก โดยเฉพาะอย่างยิ่งผู้เขียนโปรแกรมจะต้องมีความรู้เกี่ยวกับการทำงานของเครื่องเป็นอย่างดี และการที่จะทำความเข้าใจในเรื่องนี้ก็ไม่ใช่เรื่องง่าย ซึ่งรวมถึงเวลาที่ต้องใช้ในการศึกษา อันจะต้องเสียไปโดยไม่จำเป็นนัก

ถ้าเราลองหันมาดูในเรื่องของภาษา (programming language) ที่ใช้กันอยู่ก็จะเห็นได้ว่า ภาษาที่สามารถใช้ในการเขียนโปรแกรมสั่งงานเครื่องคอมพิวเตอร์อาจแบ่งออกได้เป็น 3 ชนิดได้แก่ ภาษาเครื่อง(Machine language) ภาษาแอสเซมบลี(Assembly language)และภาษาระดับสูง(High level language) ซึ่งภาษาทั้งสามชนิดนี้มีโครงสร้างของภาษาที่แตกต่างกัน ก็ลองมาดูกันว่าภาษาแต่ละชนิดนั้นมีความแตกต่างกันอย่างไรบ้าง

ภาษาเครื่อง เป็นภาษาที่ประกอบด้วยตัวเลขฐานสอง(0 และ 1) ดังนั้นชุดคำสั่งของภาษาเครื่องจึงประกอบด้วยชุดของตัวเลข 0 และ 1 ที่แตกต่างกันออกไป ตามแต่ชนิดของเครื่องคอมพิวเตอร์ เช่น เครื่องคอมพิวเตอร์อาจมีคำสั่งให้หยุด(HALT) เป็น

```
010000000000
```

ซึ่งจะเห็นได้ว่าการเขียนโปรแกรมสั่งงานเครื่องด้วยภาษานี้นั้นเป็นเรื่องยุ่งยากแค่ไหน และโอกาสที่จะมีข้อผิดพลาดในโปรแกรมที่

เขียนก็มีอยู่มาก ในขณะที่การแก้ไขจะทำได้ลำบากมากด้วย

ภาษาแอสเซมบลี เป็นภาษาที่พัฒนาจากภาษาเครื่อง โดยได้เปลี่ยนมาใช้รหัสที่เป็นตัวอักษรเขียนแทนคำสั่งเครื่องที่เป็นตัวเลข เพื่อให้สามารถเขียนโปรแกรมได้ง่ายขึ้น เช่นคำสั่ง (HALT) ก็ใช้คำสั่งเช่น

HLT

แทนคำสั่งที่เป็นตัวเลขตามตัวอย่างข้างบน ซึ่งรหัสตัวอักษรนี้จะถูกแปลให้เป็นภาษาเครื่องก่อนที่จะทำงานตามคำสั่งต่อไป อย่างไรก็ตามการเขียนโปรแกรมด้วยภาษานี้ก็ยังใช้ขั้นตอนและวิธีการเช่นเดียวกันกับภาษาเครื่อง ซึ่งทำให้การเขียนโปรแกรมยังคงมีความยุ่งยากเช่นกัน

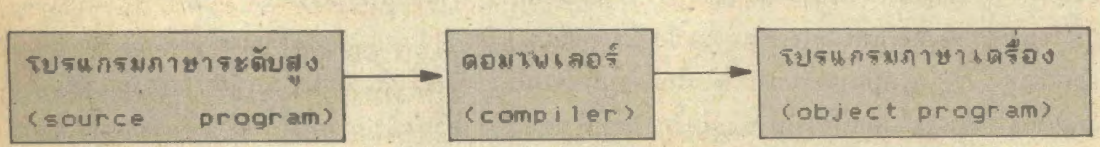
ภาษาระดับสูง ซึ่งถือว่าเป็นภาษาที่มีความเป็นอิสระจากเครื่องคอมพิวเตอร์(Machine-independent) ทำให้สามารถที่จะใช้ภาษาในระดับนี้กับเครื่องคอมพิวเตอร์ที่สร้างจากที่ต่างกันหรือต่างชนิดกันได้(ต่างระบบกัน) ซึ่งจะเห็นว่าแตกต่างจากภาษาระดับต่ำ ที่เมื่อใช้เขียนโปรแกรมคอมพิวเตอร์เครื่องหนึ่งแล้วอาจต้องเขียนใหม่ทั้งหมดเมื่อนำไปใช้กับคอมพิวเตอร์เครื่องอื่นที่มีโปรเซสเซอร์ต่างกัน โปรแกรมที่เขียนด้วยภาษาระดับสูงนั้น จะต้องถูกแปลให้เป็นรหัสภาษาเครื่องก่อนที่จะทำงานตามคำสั่งในโปรแกรมนั้น

รู้จักกับ "คอมพายเลอร์"

คราวนี้ก็มาถึงคำถามที่ว่า "คอมพายเลอร์" คืออะไรกันแน่ ซึ่งก็เป็นคำถามที่จะนำไปสู่ "บางอ้อ" กันเสียที ความจริงแล้วคอมพายเลอร์ก็เป็นซอฟท์แวร์ประเภทหนึ่งนั่นเอง โดยจะเป็นโปรแกรมที่ถูกเขียนขึ้นมาเพื่อทำหน้าที่สื่อความต้องการของโปรแกรมเมอร์ให้เครื่องคอมพิวเตอร์สามารถเข้าใจได้ โดยที่โปรแกรมเมอร์สามารถเขียนโปรแกรมด้วยภาษาระดับสูงภาษาใดภาษาหนึ่ง ที่ง่ายต่อความเข้าใจในทัศนะของโปรแกรมเมอร์ (ซึ่งเครื่องคอมพิวเตอร์ไม่สามารถเข้าใจได้โดยตรง) หรือพูดให้กระชับขึ้นอีกนิดหนึ่งก็ได้ว่า คอมพายเลอร์เป็นโปรแกรมที่ทำหน้าที่แปลภาษาระดับสูงให้เป็นภาษาเครื่องนั่นเอง โดยที่แต่ละคอมพายเลอร์ก็จะทำหน้าที่ในการแปลภาษาระดับสูงภาษาใดภาษาหนึ่งเท่านั้น เช่น FORTRAN compiler ก็จะทำหน้าที่ในการแปลภาษาระดับสูงที่เขียนขึ้น(source program) ตามกฎเกณฑ์ของภาษาฟอร์แทรนเท่านั้น จะไม่สามารถนำไปทำการแปลภาษาระดับสูงอื่น ๆ เช่น โคบอล ปาสคาลหรือภาษาอื่น ๆ ได้ ซึ่งหน้าที่ของคอมพายเลอร์นั้นจะเป็นไปตามที่ได้แสดงไว้ในรูปที่ 1

จะเห็นได้ว่าตามรูปที่ 1 นั้น โปรแกรมภาษาระดับสูงที่โปรแกรมเมอร์เขียนจะเป็นโปรแกรมที่เขียนโดยใช้คำสั่งตามกฎเกณฑ์ที่ได้ออกแบบไว้สำหรับแต่ละภาษา เช่น ภาษาฟอร์แทรน โคบอล ปาสคาล ฯลฯ โดยคอมพายเลอร์ของภาษาต่าง ๆ เหล่านี้ ก็จะทำการแปลโปรแกรมที่เขียนขึ้นด้วยภาษาดังกล่าวให้เป็นภาษาเครื่อง ซึ่งเครื่องคอมพิวเตอร์สามารถเข้าใจได้ และในปัจจุบันก็มีคอมพายเลอร์ที่ใช้สำหรับแปลภาษาต่าง ๆ อยู่มากมาย

พอถึงตรงนี้ก็อาจมีผู้สงสัยว่า ทำไมจะต้องสร้างภาษาระดับสูงขึ้นมาตั้งมากมายจนผู้ใช้ไม่รู้ว่าควรจะเลือกใช้ภาษาไหนดี มีแค่ภาษาเดียวหรือแค่สองสามภาษาไม่ได้หรือจะได้ไม่ต้องเสียเวลาเพื่อเรียนรู้วิธีการใช้(กฎเกณฑ์ของภาษา) หรือคอยติดตามนำภาษาใหม่ ๆ มาใช้ ซึ่งเหตุผลในการที่ได้มีผู้พยายามเขียนคอมพายเลอร์สำหรับภาษาระดับสูงต่าง ๆ ขึ้นมาใช้มากมายหลายภาษานั้น เนื่องจากปัญหาที่เกิดจากความยุ่งยากในการติดต่อหรือสั่งให้เครื่องคอมพิวเตอร์ทำงานด้วยภาษาเครื่อง หรือภาษาแอสเซมบลีอย่างที่ได้คุยกันมาแล้ว และความต้องการใช้งานเครื่องนั้นก็มีความแตกต่างกันอย่างมาก



รูปที่ 1 แสดงหน้าที่ของคอมพายเลอร์

ฟอร์แทรน → คำนวณ
พีแอลวัน → ธุรกิจ
อาร์พีจี → ธุรกิจ
โคบอล → บัญชี
อัลกอ → คำนวณ
เบสิก → ทั่วไป
ซี → ทั่วไป

แต่ละกลุ่มของผู้ใช้ เช่นผู้ใช้บางกลุ่มอาจต้องการใช้งานในทางการคำนวณ ในขณะที่บางกลุ่มอาจต้องการใช้งานในทางธุรกิจ ซึ่งมีการคำนวณน้อย หรือใช้ในงานด้านเอกสารที่ไม่มีการคำนวณเลย หรือใช้ในงานที่ประกอบด้วยงานหลาย ๆ ประเภทรวมกัน เป็นต้น รวมถึงการขยายตัวในการนำเครื่องคอมพิวเตอร์มาใช้ในงานต่าง ๆ มากขึ้น จากสาเหตุต่าง ๆ เหล่านี้ทำให้เกิดผลที่ตามมาก็คือความหลากหลายของภาษาในระดับสูง เช่น ฟอร์แทรน โคบอล อัลกอ อาร์พีจี พีแอลวัน ลิสป์ ปาสคาล อดา เบสิก ซี รวมทั้งยังมีอีกหลายภาษาที่อยู่ในระหว่างการพัฒนา เช่น MODULA และอาจมีภาษาอื่น ๆ อีกในอนาคตเช่นภาษารุ่นที่ 4 (Fourth generation language) อย่างไรก็ตามภาษาระดับสูงนั้นไม่ได้เป็นที่นิยมนำมาใช้อย่างเท่าเทียมกัน โดยมีเพียงไม่กี่ภาษาเท่านั้นที่เป็นที่นิยมใช้กันโดยทั่วไป ซึ่งภาษาเหล่านั้นได้แก่ ฟอร์แทรน โคบอล พีแอลวัน อาร์พีจี ปาสคาล และเบสิก เป็นต้น การที่ภาษาอะไรจะเป็นที่นิยมมากน้อยแค่ไหนนั้น ย่อมขึ้นอยู่กับความยากง่ายในการทำความเข้าใจเพื่อใช้ภาษานั้น ๆ ตลอดจนขีดความสามารถของภาษาในการนำมาใช้สั่งงานเครื่องคอมพิวเตอร์ และลักษณะงานที่ผู้ใช้เครื่องคอมพิวเตอร์ต้องการ เพราะภาษาแต่ละภาษาต่างก็มีจุดเด่นที่แตกต่างกัน เช่น

ภาษา FORTRAN (FORmular TRANslation) เป็นภาษาที่นิยมใช้กันมากในอเมริกา เป็นภาษาที่เหมาะกับงานด้านการคำนวณทางวิทยาศาสตร์และวิศวกรรมศาสตร์ รวมทั้งงานด้านการค้นคว้าวิจัยมีการคำนวณทางด้านสถิติ เช่น Multiple regression, Factor analysis หรือการแก้สมการต่าง ๆ เช่น Linear equations เป็นต้น

ภาษา ALGOL(ALGOrithmic Language หรือ ALGebraic Oriented Language) เป็นภาษาที่ได้มีการพัฒนามาจากภาษา IAL (International Algebraic Language) ALGOL เป็นภาษาที่นิยมใช้กันในทวีปยุโรป เป็นภาษาที่เหมาะกับงานทางด้านการคำนวณต่าง ๆ เช่น คณิตศาสตร์ วิศวกรรมศาสตร์ และสถิติ เป็นต้น

ภาษา RPG (Report Program Generator) เป็นภาษาที่เหมาะสมกับงานด้านธุรกิจที่มีการทำรายงานต่าง ๆ หลายรูปแบบ เพราะสามารถจัดทำรูปแบบรายงานในลักษณะต่าง ๆ ได้ง่ายกว่าภาษาอื่น

ภาษา COBOL (Common Business Oriented Language) เป็นภาษาที่นำมาใช้ตั้งแต่ปี 1961 เป็นต้นมา โดยได้เปลี่ยนชื่อมาจากภาษา CODASYL (Conference Data Systems Language) เหมาะกับงานประเภท file processing เช่น งานด้านบัญชี การทำ payroll หรืองานอื่น ๆ ที่มีการคำนวณไม่มากนัก

ภาษา PL/I (Programming Language one) เป็นภาษาที่ได้รับการพัฒนาเพื่อใช้ในงานด้านธุรกิจและวิทยาศาสตร์ แต่เป็นภาษาที่ไม่ได้เน้นหนักในด้านการคำนวณเหมือน FORTRAN หรือ ALGOL และไม่ได้เน้นหนักในด้านธุรกิจเช่นเดียวกับ COBOL

ภาษา C เป็นภาษาที่พัฒนาจากหลักการของภาษา B เป็นภาษาใหม่ที่ได้มีการพัฒนาขึ้นใช้ในปี 1972 ภาษานี้มีความคล่องตัวในการประยุกต์ใช้กับงานต่าง ๆ ได้อย่างดีโดยมีโครงสร้างแบบภาษาระดับสูง และสามารถใช้งานได้รวดเร็วเหมือนภาษาระดับต่ำ เป็นภาษาที่มีการนิยมใช้อย่างมากตั้งแต่คอมพิวเตอร์ขนาดเล็กเช่นไมโครคอมพิวเตอร์ จนถึงเครื่องขนาดใหญ่เช่น เมนเฟรม

ภาษา BASIC (Beginner's Allpurpose Symbolic Institution Code) เป็นภาษาที่ถูกนำมาใช้ในกลางปี 1960 โดย Kemeny และ Kurtz แห่งมหาวิทยาลัย Dartmatz แห่งสหรัฐอเมริกา ภาษานี้ได้รับการพัฒนาเรื่อยมาจนกระทั่งในปี 1975 ได้มีการพัฒนาให้สามารถใช้กับเครื่องคอมพิวเตอร์ขนาดเล็กได้ (ไมโครคอมพิวเตอร์) เป็นภาษาที่ง่ายต่อการทำความเข้าใจ และใช้งานในลักษณะต่าง ๆ ได้อย่างกว้างขวางพอควร

นอกจากความแตกต่างกันของขีดความสามารถในการทำงานของแต่ละภาษา ที่ทำให้ได้รับความนิยมมากน้อยต่างกันแล้ว เหตุผลที่สำคัญอีกอย่างหนึ่งที่จะลืมไม่ได้ก็คือ ขีดความสามารถของคอมพายเลอร์ของภาษาต่าง ๆ เหล่านั้นด้วย เพราะคอมพายเลอร์ที่มีความต้องการทางด้านฮาร์ดแวร์น้อย มีความสามารถทำงานได้ รวดเร็ว มีประสิทธิภาพ และติดต่อสื่อสารกับผู้ใช้ด้วยภาษาและ

ปัจจุบันภาษาคอมพิวเตอร์ได้รับการพัฒนาขึ้นสำหรับงานแต่ละประเภท และขณะเดียวกัน คอมพายเลอร์ ก็จำเป็นสำหรับความสำเร็จของภาษาเหล่านั้น

ขั้นตอนที่ง่ายต่อการทำความเข้าใจ ก็จะมีส่วนในการสร้างความนิยมให้กับโปรแกรมเมอร์ด้วยเช่นกัน

ทำไมต้อง "คอมพายเลอร์"

หัวข้อต่อไปที่จะนำมาคุยกันก็คือ คำถามที่ว่า การเขียนคอมพายเลอร์นั้น เขียนขึ้นมาทำไม มีวัตถุประสงค์อย่างไร ซึ่งในเรื่องวัตถุประสงค์ของการเขียนคอมพายเลอร์นั้นที่เห็นได้เด่นชัดในทันทีที่พูดถึงก็คือ เพื่อใช้แปลภาษาระดับสูงให้เป็นภาษาระดับต่ำที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ เนื่องจากภาษาระดับสูงเป็นภาษาที่คนเราสามารถทำความเข้าใจได้ง่าย แต่เครื่องคอมพิวเตอร์ไม่สามารถเข้าใจได้ จึงต้องใช้คอมพายเลอร์ทำหน้าที่แปลความและเปลี่ยนให้เป็นภาษาเครื่องก่อน จึงจะสามารถให้เครื่องคอมพิวเตอร์ทำงานตามขั้นตอนที่กำหนดไว้ในโปรแกรมได้ หรืออาจพูดได้ว่า คอมพายเลอร์จะถูกใช้เป็นสื่อกลางในการติดต่อสื่อสารกันระหว่างใช้เป็นสื่อกลางในการติดต่อสื่อสารกันระหว่างผู้เขียนโปรแกรม(ในภาษาระดับสูง) communication) ซึ่งหมายถึงการใช้คอมพายเลอร์เป็นล่ามให้โปรแกรมเมอร์สามารถติดต่อสื่อสารกับคอมพิวเตอร์ได้

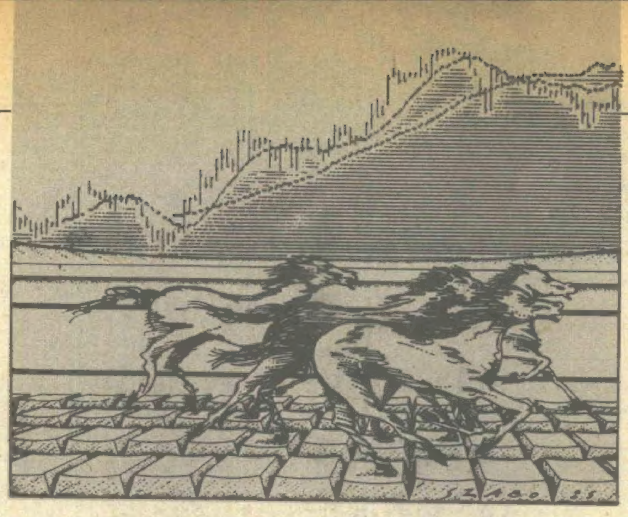
วัตถุประสงค์อีกประการหนึ่งก็ได้แก่ ความพยายามที่จะพัฒนาขีดความสามารถในการสั่งงานเครื่องคอมพิวเตอร์ โดยการพัฒนาภาษาที่ผู้เขียนโปรแกรมจะได้รับความสะดวกในขณะที่สามารถสั่งให้เครื่องทำงานได้อย่างมีประสิทธิภาพ และสนองความต้องการของผู้เขียนโปรแกรมได้มากขึ้นด้วย ซึ่งหมายถึงความต้องการที่จะให้มีคอมพายเลอร์ที่มีประสิทธิภาพ และสามารถสร้างคำสั่งในระดับภาษาเครื่อง(โดยแปลจากภาษาระดับสูงที่กำหนดไว้) ที่มีขบวนการในการประมวลผลได้อย่างถูกต้อง รวดเร็ว และใช้ทรัพยากร (Hardware resource) น้อยที่สุด

นอกจากนี้ยังมีวัตถุประสงค์ที่จะแก้ปัญหาที่เกิดจากความยุ่งยากซับซ้อนของการเขียนโปรแกรมในภาษาระดับต่ำ ซึ่งมีปัญหาทั้งในด้านการสื่อความหมายของภาษาที่ใช้ และขั้นตอนการเขียนคำสั่งที่ต้องเป็นไปตามขบวนการการทำงานของเครื่อง และยังรวมถึงความยุ่งยาก และเวลาที่ต้องสูญเสียไปกับการศึกษาขั้นตอนและคำสั่งการทำงานของเครื่องซึ่งไม่ใช่เรื่องจำเป็นนัก ถ้าหากมีภาษาระดับสูงที่ดีซึ่งสามารถสั่งงาน(เขียนโปรแกรม) ได้ง่ายและมีโครงสร้างที่ดีรวมถึงการมีคอมพายเลอร์ความสามารถสูงด้วย (คอมพายเลอร์ซึ่งสามารถแปลภาษาที่ใช้เขียนโปรแกรมให้เป็นภาษาเครื่อง และสามารถนำไปประมวลผลได้อย่างมีประสิทธิภาพ) วัตถุประสงค์อีกอย่างหนึ่งก็คือ การหาเครื่องมือ (ในที่นี้หมายถึงคอมพายเลอร์) ที่จะใช้สนับสนุนการเขียนโปรแกรมสั่งงานเครื่องคอมพิวเตอร์ ให้เป็นไปตามความต้องการของโปรแกรมเมอร์ได้มากที่สุดและรวดเร็วที่สุดด้วย (เทียบเท่าหรือใกล้เคียงกับการทำงานที่เขียนโปรแกรมด้วยภาษาระดับต่ำ) และจากสาเหตุที่ผู้ใช้มีความต้องการใช้เครื่องคอมพิวเตอร์ในงานด้านต่าง ๆ ผิดแผกแตกต่างกันออกไปมากมาย เพราะฉะนั้นจึงนับเป็นเหตุผลอีกประการหนึ่งที่จำเป็นต้องมีภาษาที่เหมาะสมกับงานแต่ละประเภท และการที่ภาษาต่าง ๆ ซึ่งสร้างขึ้นนั้นจะประสบความสำเร็จได้ คือ มีผู้นิยมและนำไปใช้อย่างแพร่หลาย ก็ต่อเมื่อมีการเขียนคอมพายเลอร์เพื่อทำหน้าที่ในการแปลความของแต่ละภาษาเหล่านั้นด้วย

มาดูคอมพายเลอร์ทำงานกันดีกว่า

ตามที่ได้พูดไว้ในตอนต้นแล้วว่า หน้าที่หลักของคอมพายเลอร์นั้นคือ การแปลภาษาระดับสูงที่โปรแกรมเมอร์เขียนขึ้น ให้เป็นภาษาที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ ทีนี้ลองมาดูกันว่า คอมพายเลอร์นั้นทำงานอย่างไร หรือมีขั้นตอนในการแปลภาษาต่าง ๆ ได้อย่างไร

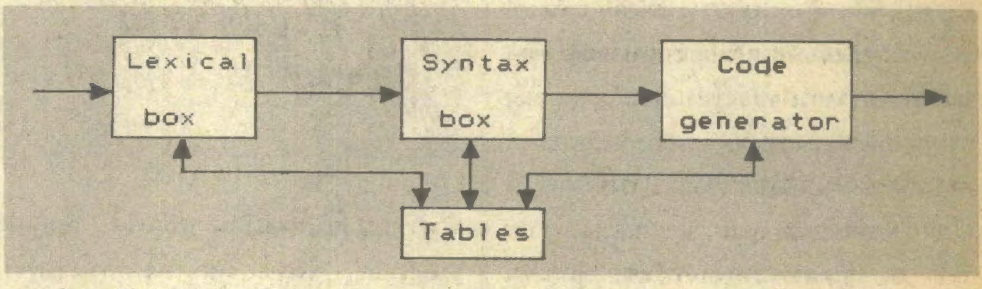
ขบวนการในการทำงานของคอมพายเลอร์นั้น มีขั้นตอนที่ค่อนข้างจะยุ่งยากซับซ้อน ซึ่งจะมีมากน้อยแค่ไหนก็ขึ้นอยู่กับขีดความสามารถของภาษาที่ผู้ออกแบบต้องการ ถ้าออกแบบให้มีขีดความสามารถสูงความสลับซับซ้อนของขั้นตอนการทำงานของคอมพายเลอร์ก็จะมีมากขึ้นตามไปด้วย แต่ไม่ว่าจะมีความซับซ้อนมากน้อยแค่ไหน ขบวนการในการทำงานดังกล่าวก็จะต้องครอบคลุมตั้งแต่ขั้นตอนของการอ่าน และตรวจสอบความถูกต้องของคำสั่งจากภาษาที่กำหนดขึ้นว่าเป็นไปตามกฎเกณฑ์ที่ได้กำหนดไว้หรือเปล่า จนถึงสิ้นสุดขั้นตอนการแปลเป็น



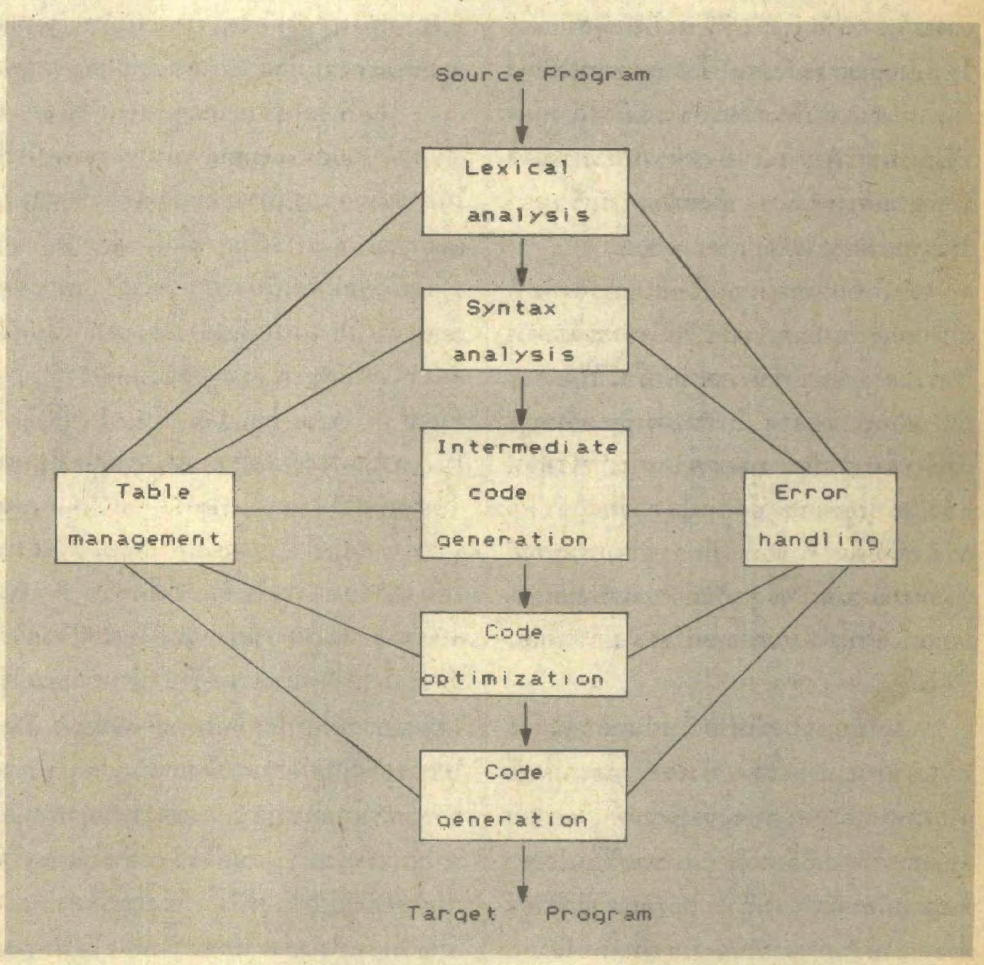
รหัสภาษาเครื่องที่มีความหมายหรือทำงานได้ผลลัพธ์เช่นเดียวกันกับโปรแกรมที่เขียนขึ้น ซึ่งอาจมีขั้นตอนการทำงานแบบง่าย ๆ ตามที่ได้แสดงไว้ในรูปที่ 2

ขั้นตอนต่าง ๆ ในระหว่างการแปล (ขบวนการทำงานของคอมพายเลอร์) นั้นอาจแตกต่างกันออกไปตามความต้องการของผู้ออกแบบ ซึ่งจำเป็นจะต้องพิจารณาถึงความเหมาะสม และประสิทธิภาพในการทำงาน(ระหว่างขบวนการแปรรหัส) ของคอมพายเลอร์ นอกจากนี้ยังต้องรวมถึงการอำนวยความสะดวกให้กับโปรแกรมเมอร์ ในด้านข่าวสารความคลาดเคลื่อนของโปรแกรมที่เขียนขึ้น(ถ้ามี) หรืออาจรวมถึงขั้นตอนอื่น ๆ ที่จะอำนวยประโยชน์ให้กับผู้ใช้คอมพายเลอร์ด้วย ถึงแม้ว่าขั้นตอนของขบวนการทำงานของคอมพายเลอร์จะแตกต่างกันออกไปด้วยเหตุผลต่าง ๆ หลายประการก็ตาม แต่ขั้นตอนที่คอมพายเลอร์ควรจะมีนั้น ประกอบด้วยขั้นตอนต่าง ๆ 6 ขั้นตอน(ตามรูปที่ 3) ด้วยกัน ได้แก่

1. ขั้นตอนในการอ่านคำสั่งจากโปรแกรมที่เขียนขึ้นด้วยภาษาระดับสูง เพื่อทำการจัดแบ่งคำสั่งต่าง ๆ เหล่านั้นเป็นกลุ่ม ๆ (ที่ถูกจะเรียกว่า token) ตามลักษณะและหน้าที่ตามที่ได้ระบุไว้ในโครงสร้างและกฎเกณฑ์ของภาษาได้แก่ key word(เช่น DO, IF....) หรือ identifier ต่างๆ (เช่น X, NUM, ...) หรือสัญลักษณ์ของเครื่องหมายทางคณิตศาสตร์(เช่น < =, + ...) และสัญลักษณ์อื่นๆ (เช่น (,), comma....) การ



รูปที่ 2 ขั้นตอนการทำงานแบบง่ายๆ ของคอมพายเลอร์



รูปที่ 3 ขบวนการการทำงานของคอมพายเลอร์

ทำงานในขั้นตอนนี้เรียกว่า Lexical analysis

2. ขั้นตอนการตรวจสอบความถูกต้องของโครงสร้างคำสั่งของโปรแกรม ว่าถูกต้องตามกฎเกณฑ์ของภาษา(syntactic structures) ตามที่ได้กำหนดไว้หรือเปล่า ในขั้นตอนนี้เรียกว่า analysis

3. ขั้นตอนการสร้างรหัสของชุดคำสั่งที่ผ่านการตรวจสอบความถูกต้องของโครงสร้างแล้ว ให้อยู่ในรูปแบบหรือขั้นตอนการทำงานของเครื่อง รหัสที่ได้นี้เป็นรหัสที่เรียกว่า intermediate code ซึ่งอาจเป็นรหัสของภาษาแอสเซมบลี แต่จะยังไม่มีการกำหนดรีจิสเตอร์ต่างๆ ที่จะใช้ในการประมวลผลให้กับรหัสเหล่านี้ ซึ่งขั้นตอนนี้เรียกว่า intermediate code generation

4. ขั้นตอนการแก้ไขปรับปรุงรหัสที่เป็น intermediate code ให้มีประสิทธิภาพในการทำงานมากขึ้น และใช้เนื้อที่ในหน่วยความจำหลักน้อยลง ในขั้นตอนนี้จะเป็นการกำจัดชุดคำสั่งที่ทำงานซ้ำซ้อนกัน หรือชุดของ expression ย่อยที่มีการเรียกใช้บ่อยครั้ง หรืออื่นๆ ที่ จะทำให้บรรลุวัตถุประสงค์ข้างต้น ขั้นตอนนี้จะถูกเรียกว่า code optimization ซึ่งบางคอมพายเลอร์อาจไม่มีขั้นตอนนี้ก็ได้

5. ขั้นตอนสุดท้ายเป็นขั้นตอนในการสร้างรหัสภาษาเครื่อง ซึ่งในขั้นตอนนี้จะมีการกำหนดตำแหน่งของข้อมูลในหน่วยความจำ และทำการเลือกรหัสในการเข้าถึง(access) ข้อมูลแต่ละตัว รวมถึงการกำหนดรีจิสเตอร์ต่างๆ สำหรับการประมวลผล ขั้นตอนนี้เรียกว่า code generation และเมื่อสิ้นสุดการทำงานของขั้นตอนนี้แล้วก็จะได้ object program ซึ่งเครื่องคอมพิวเตอร์สามารถนำไปประมวลผลตามคำสั่งต่อไปได้

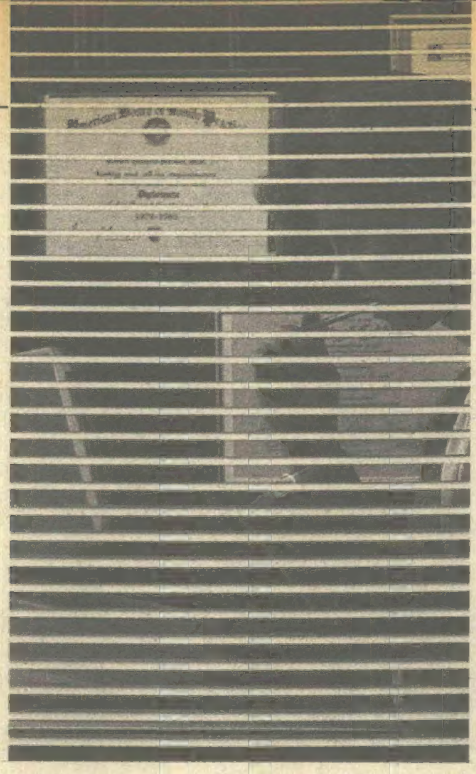
อ่านคำสั่งจากโปรแกรม
ตรวจสอบความถูกต้อง
สร้างรหัสของชุดคำสั่ง
ปรับปรุงรหัสให้ดีขึ้น
สร้างรหัสภาษาเครื่อง

ห้าขั้นห้าตอน
ที่คอมพายเลอร์ควรมี

ในการทำงานของคอมพายเลอร์ทุกๆ ขั้นตอนนั้น จะต้องทำงานร่วมกับ table management และ Error handling ตลอดเวลา โดยที่ table management หรืออาจเรียกว่า bookkeeping นั้น จะทำหน้าที่ในการเก็บรวบรวมรายละเอียดของชื่อต่างๆ (names) ที่มีการเรียกใช้ในโปรแกรม ซึ่งรายละเอียดที่เก็บนั้นจะช่วยให้รู้ว่า แต่ละชื่อคืออะไร มีคุณสมบัติอย่างไร เช่น เป็นตัวแปรชนิด integer, real หรือเป็นค่าคงที่ เป็นต้น ส่วน error handler นั้น จะทำหน้าที่ในการส่งผ่านข่าวสารความคลาดเคลื่อนที่เกิดขึ้น(ถ้ามี) ของโปรแกรม ไปยังขั้นตอนการทำงานถัดไปของคอมพายเลอร์ซึ่งจะทำให้สามารถทำงานในขั้นตอนต่อไปได้อย่างต่อเนื่อง นอกจากนี้ยังรวมถึงหน้าที่ในการนำข่าวสารดังกล่าว แจ้งให้โปรแกรมเมอร์ได้รู้ถึงความคลาดเคลื่อนต่างๆ เหล่านั้นด้วย ในการออกแบบคอมพายเลอร์นั้น ผู้ออกแบบอาจกำหนดให้ขบวนการทำงานของคอมพายเลอร์ตามขั้นตอนต่างๆ เหล่านี้ ทำเสร็จสิ้นในครั้งเดียวเลยก็ได้ ซึ่งหมายความว่าเมื่อเริ่มต้นอ่านประโยคคำสั่งของโปรแกรม และทำงานตามขั้นตอนของ lexical analysis แล้วก็จะทำงานตามขั้นตอนอื่นๆ ต่อไปจนเสร็จสิ้นขั้นตอนของ code generation แล้วจึงกลับไปอ่านคำสั่งต่อๆ ไปในลักษณะเดียวกันจนจบโปรแกรม ในทำนองเดียวกัน ผู้ออกแบบอาจกำหนดให้มีการทำงานเป็นช่วงๆ ตามการทำงานในแต่ละขั้นตอน หรือรวมกัน 2–3 ขั้นตอน(หรือมากกว่า) ให้เสร็จเรียบร้อยหมดทั้งโปรแกรมก่อน แล้วจึงนำเอาท์พุทที่ได้(ซึ่งอาจเก็บไว้ใน intermediate file) ไปเป็นอินพุทของขั้นตอนถัดไปก็ได้ ซึ่งช่วงการทำงานในแต่ละช่วงเช่นนี้เรียกว่า "pass" จำนวนของ pass ที่ใช้ในคอมพายเลอร์นั้นจะมีผลต่อจำนวนหน่วยความจำที่ต้องการใช้และความเร็วในการทำงานของคอมพายเลอร์ โดยที่คอมพายเลอร์ซึ่งมีจำนวน pass น้อยกว่า จะสามารถทำงานได้เร็วกว่าแต่ต้องการใช้หน่วยความจำหลักเป็นจำนวนมากกว่า ในทางกลับกัน การให้มีจำนวน pass น้อยลงจะทำให้ประหยัดเนื้อที่ในหน่วยความจำหลักได้มากขึ้น แต่การทำงานของคอมพายเลอร์ก็จะช้าลง เพราะฉะนั้นการที่ผู้ออกแบบคอมพายเลอร์จะใช้วิธีการแบบไหนก็ย่อมต้องพิจารณาถึงข้อจำกัดของฮาร์ดแวร์ที่ใช้ โครงสร้างของภาษา และความเร็วในการทำงานของคอมพายเลอร์ รวมถึงเรื่องอื่นๆ เช่นความรู้ ความสามารถของผู้เขียนคอมพายเลอร์เอง หรือเวลาที่มีตลอดจนงบประมาณที่ต้องใช้ในการเรียนคอมพายเลอร์ขึ้นมา

ภาษาเขียนคอมพายเลอร์นั้นเป็นไฉน

ทีนี้ลองมาดูกันว่าการเขียนโปรแกรม

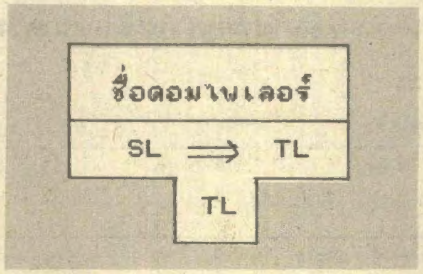


คอมพายเลอร์นั้น จะใช้ภาษาอะไรดี หรือภาษาที่เหมาะสมที่จะนำมาใช้ในการเขียนนั้นคือภาษาอะไรบ้าง หรือมีภาษาอะไรที่เป็นภาษาเฉพาะที่จะใช้ในการเขียนโปรแกรมคอมพายเลอร์หรือเปล่า

การเขียนโปรแกรมคอมพายเลอร์นั้นสามารถเขียนด้วยภาษาคอมพิวเตอร์ได้เกือบทุกภาษา แต่ภาษาที่เหมาะสมกับการเขียนโปรแกรม(implementation languages)ชนิดนี้ ควรเป็นภาษาที่สามารถเขียนได้ในลักษณะของโปรแกรมโครงสร้าง (structured programming techniques) เพราะภาษาที่มีคุณสมบัติแบบนี้จะช่วยให้การเขียนโปรแกรม การตรวจสอบ และการปรับปรุงแก้ไข ทำได้สะดวก ภาษาที่นิยมนำมาใช้เขียนโปรแกรมคอมพายเลอร์มีอยู่หลายภาษาได้แก่ FSL, XPL, CDL และภาษาโครงสร้างต่างๆ เช่นภาษาปาสคาล, ALGOL68 และภาษา C เป็นต้น ฉะนั้นการที่ผู้เขียนโปรแกรมคอมพายเลอร์จะเลือกใช้ภาษาใด ก็ย่อมขึ้นอยู่กับเหตุผลหลายประการเช่นความสะดวกในการเลือกใช้ภาษา ความถนัดของผู้เขียน และคุณภาพของคอมพายเลอร์ที่ต้องการเป็นต้น แต่การที่จะเขียนคอมพายเลอร์ได้ดีแค่ไหนนั้น ก็ขึ้นอยู่กับความรู้ความสามารถของผู้เขียนเกี่ยวกับภาษาที่เลือกใช้ด้วย เพราะอาจเป็นไปได้ที่ภาษาซึ่งผู้เขียนมีความถนัดอาจไม่เหมาะกับการเขียนคอมพายเลอร์ที่ออกแบบไว้ จึงต้องใช้ภาษาอื่นที่อาจไม่มีความถนัด จึงอาจจำเป็นจะต้องศึกษาหาความรู้เพิ่มเติมเพื่อให้สามารถเขียนโปรแกรมคอมพายเลอร์ที่มีคุณภาพตามต้องการได้

ภาษาคอมพิวเตอร์ที่เกี่ยวข้องกับการเขียนคอมพายเลอร์นั้น อาจแบ่งออกเป็นกลุ่มใหญ่ๆ ได้ 3 ภาษา ซึ่งก็ได้แก่

– ภาษาที่โปรแกรมเมอร์ใช้เขียนโปร–แกรม(ไม่ใช่ภาษาเครื่อง) เรียกภาษานี้ว่า source language (SL)



รูปที่ 4 T ไดอะแกรมแสดงความสัมพันธ์ของภาษาต่างๆ

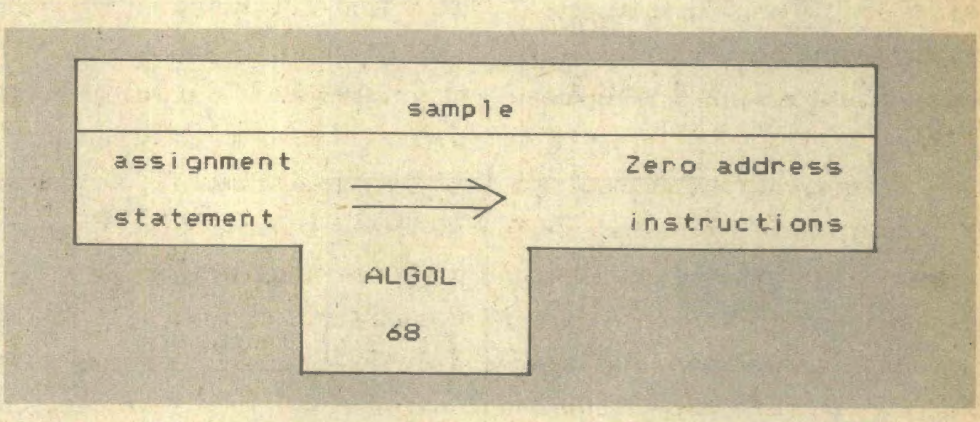
– ภาษาที่ใช้ในการเขียนโปรแกรมคอมพายเลอร์ ซึ่งอาจเป็นภาษาใดๆ ก็ได้ตามที่ได้พูดไว้ก่อนหน้านี้แล้ว ภาษานี้เรียกว่า implementation language (IL)

– ภาษาที่ได้หลังจากการแปล source language โดยคอมพายเลอร์แล้ว ซึ่งเรียกภาษานี้ว่า target language (TL) ซึ่งภาษาที่ได้จะเป็นภาษาที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้

สำหรับความเกี่ยวข้องสัมพันธ์กันของภาษาต่างๆ เหล่านี้ อาจแสดงในรูปแบบของ T ไดอะแกรมตามที่ได้แสดงไว้ในรูปที่ 4

ในที่นี้ถ้าสมมุติว่ากำหนดให้ใช้ภาษา ALGOL-68 ในการเขียนคอมพายเลอร์ (ใช้ภาษา ALGOL-68 เป็น implementation language) และให้คอมพายเลอร์ตัวนี้มีชื่อว่า Sample แล้ว ในรูปที่ 4 ก็จะเปลี่ยนเป็นรูปตามที่ได้แสดงไว้ในรูปที่ 5

ภาษาที่เกี่ยวข้องในการเขียนคอม–



รูปที่ 5 T ไดอะแกรมที่ใช้ภาษา ALGOL-68

ภาษาทั้ง 3 ที่เกี่ยวข้องกับการเขียนคอมพายเลอร์

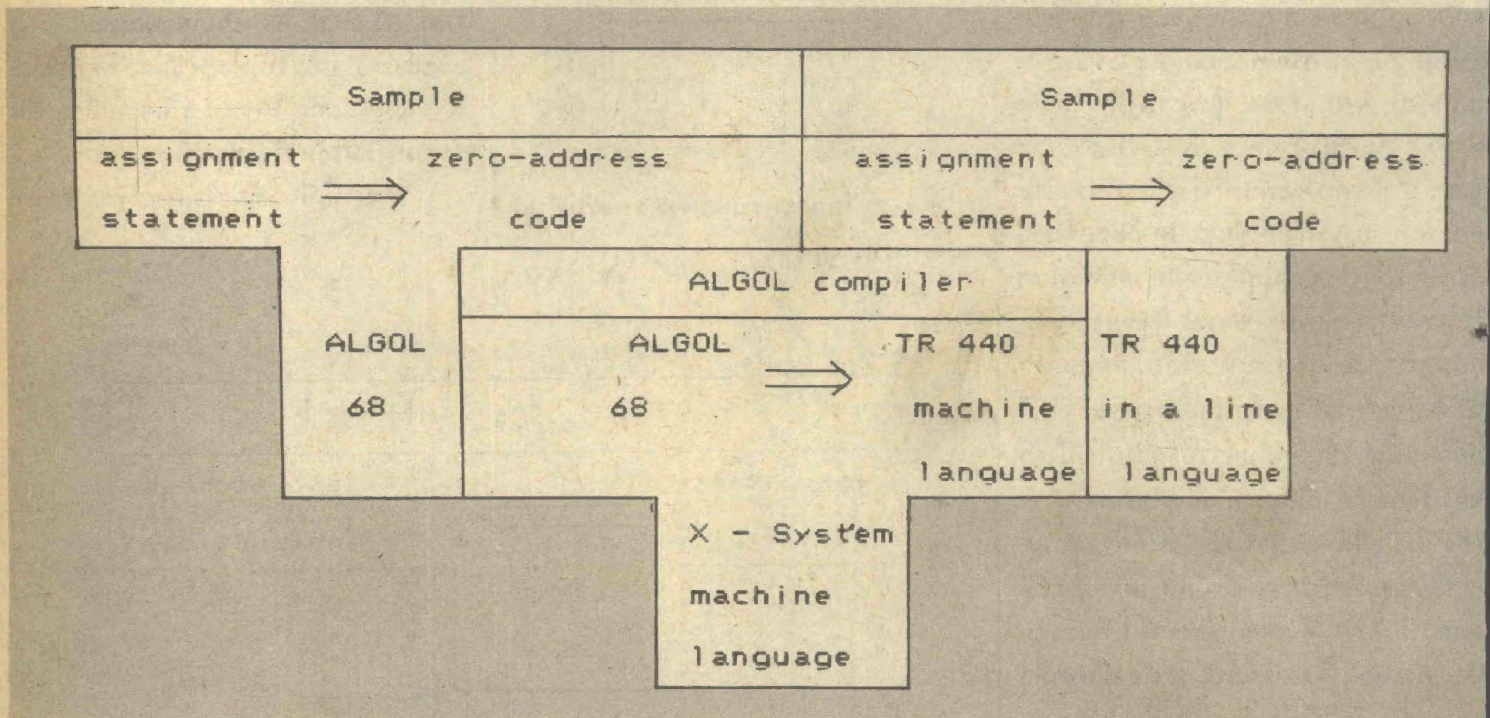
- *Source Language (SL)*
- *Implementation Language (IL)*
- *Target Language (TL)*

พายเลอร์ตามรูปที่ 5 นั้น ไม่ใช่มีเพียงภาษาที่กำหนดไว้และภาษา ALGOL-68 เท่านั้น เพราะจะต้องมีคอมพายเลอร์ของภาษา ALGOL-68 เข้ามาเกี่ยวข้องในการแปลโปรแกรมคอมพายเลอร์ที่เขียนด้วยภาษา ALGOL-68 ก่อน และถ้าสมมุติว่าเราทำการเขียนคอมพายเลอร์นี้บนเครื่อง X - System และให้คอมพายเลอร์ที่ได้ทำงานบนเครื่องนี้แล้ว ภาษาที่ได้ในครั้งสุดท้าย(TL) ก็จะเป็นรหัสของภาษาเครื่องของ X - System นั่นเอง ตามที่ได้แสดงไว้ในรูปที่ 6

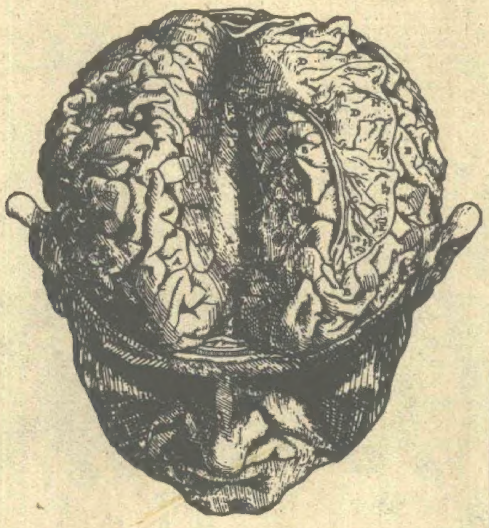
ในการเขียนคอมพายเลอร์นั้น ผู้เขียนอาจจะเขียนให้โปรแกรมคอมพายเลอร์ทำงานบนเครื่องคอมพิวเตอร์ที่ใช้เขียนเอง หรืออาจเขียนบนเครื่องคอมพิวเตอร์ระบบหนึ่งเพื่อให้สามารถทำงานได้ในเครื่องระบบอื่นก็ได้ เช่นการเขียนโปรแกรมคอมพายเลอร์ในเครื่องระดับมินิ หรือเมนเฟรม เพื่อให้คอมพายเลอร์ที่ได้นั้น สามารถนำไปใช้ในเครื่องระดับไมโคร เป็นต้น การเขียนคอม-พายเลอร์เพื่อให้สามารถใช้ได้ในเครื่องคอมพิวเตอร์หลายๆ ระบบนั้น ก็จะทำให้ได้รับประโยชน์มากขึ้น แต่ก็หมายความถึงการที่จะต้องมีภาษาที่เข้ามาเกี่ยวข้องในการเขียนนี้มากขึ้นด้วย โดยเฉพาะอย่างยิ่งภาษาระดับต่ำที่จะนำมาเป็น target language

คอมพายเลอร์จำเป็นหรือ?

ถ้าพิจารณาถึงอัตราการขยายตัวของธุรกิจคอมพิวเตอร์ ทั้งในแง่ของผู้ผลิตและผู้ใช้เครื่องคอมพิวเตอร์แล้ว ก็ไม่น่าจะมีข้อสงสัยอีกต่อไปว่า คอมพายเลอร์นั้นมีความจำเป็นต่อผู้ใช้เครื่องคอมพิวเตอร์หรือไม่ เพราะ



รูปที่ 6 T ไดอะแกรมที่เป็นรหัสของ X - System



คำตอบที่จะเป็นไปได้นั้นน่าจะมีเพียงคำตอบเดียวเท่านั้นคือ "จำเป็น" อย่างยิ่งที่จะต้องมีคอมพายเลอร์สำหรับใช้ในงานด้านนี้

ในปัจจุบันเทคโนโลยีคอมพิวเตอร์ได้มีการพัฒนาไปอย่างรวดเร็ว และมีการแข่งขันกันอย่างกว้างขวางทั่วโลก ทำให้มีการผลิตเครื่องคอมพิวเตอร์ขึ้นมามากมายหลายระบบหลายขนาด ในขณะที่ราคาของเครื่องคอมพิวเตอร์รุ่นใหม่ๆ นั้นมีราคาถูกลง แต่กลับมีขีดความสามารถสูงขึ้น ดังนั้นผลที่ตามมาก็คือการได้รับความสนใจจากนักธุรกิจ ตั้งแต่นักธุรกิจในธุรกิจขนาดเล็กไปจนถึงนักธุรกิจที่มีกิจการหรือธุรกิจครอบคลุมอยู่ทั่วโลก หรือแม้แต่องค์กรของรัฐ นักศึกษา และประชาชนทั่วไป จึงได้ทำให้มีการนำเครื่องคอมพิวเตอร์มาใช้ในวงการต่างๆ อย่างแพร่หลาย ซึ่งอาจใช้เป็นเหตุผลที่จะพูดได้ว่า การที่จะให้ผู้ใช้รวมถึงโปรแกรมเมอร์ทั้งหลายต้องมาคอยติดตาม ศึกษาชุดคำสั่งและการทำงานของเครื่องแต่ละรุ่นแต่ละระบบ ซึ่งมีการพัฒนาและเปลี่ยนแปลงอย่างรวดเร็ว(เช่นเปลี่ยนจากไมโครคอมพิวเตอร์ขนาด 8 บิท เป็น 16 บิท และ 32 บิท ในระยะเวลาเพียงไม่กี่ปี) ในขณะที่ต่างก็มีความต้องการจะใช้งานเครื่อง หรือระบบคอมพิวเตอร์ให้เร็วที่สุดนั้นเป็นสิ่งที่แทบจะเป็นไปไม่ได้

ในที่สุดก็จำเป็นต้องพึ่งพาคอมพายเลอร์ (ซึ่งถือได้ว่าเป็นพระเอกขี่ม้าขาวมาช่วย) ทำหน้าที่เป็นสื่อกลางในการใช้เครื่องคอมพิวเตอร์ทั้งหลายเหล่านั้น เพราะคุณสมบัติอย่างหนึ่งของคอมพายเลอร์ คือความสามารถในการสื่อสาร ที่จะทำให้สามารถสั่งงานเครื่องคอมพิวเตอร์ด้วยภาษาระดับสูงที่มีความเป็นอิสระต่อเครื่อง โดยไม่จำกัดว่าจะเป็นเครื่องคอมพิวเตอร์รุ่นใดหรือระบบใด ถ้าหากว่ามีคอมพายเลอร์ของภาษานั้นๆ ใช้อยู่ด้วยแล้วก็จะสามารถติดต่อสื่อสารกันได้ นอกจากนี้การที่ได้มีการพัฒนาภาษาระดับสูงอยู่อย่างต่อเนื่องเช่นในปัจจุบัน ยิ่งจะทำให้คอมพายเลอร์มีบทบาทและทวีความสำคัญในการใช้คอมพิวเตอร์มากยิ่งขึ้น

ส่งท้าย

จนถึงช่วงนี้เราก็ได้คุยกันในเรื่องของคอมพายเลอร์มาพอสังเขปอย่างน้อยก็ได้คุยกันว่า คอมพายเลอร์นั้นเป็นโปรแกรมที่เขียนขึ้นมาเพื่อให้ทำหน้าที่แปลภาษาระดับสูงที่มีความเข้าใจง่าย คือมีความใกล้เคียงกับภาษาที่ใช้อยู่ในชีวิตจริงและเป็นภาษาค่อนข้างเป็นสากล (ซึ่งได้แก่ภาษาอังกฤษและทางภาษาคณิตศาสตร์.....) รวมถึงโครงสร้างของภาษาที่เป็นธรรมชาติ ทั้งในด้านการตัดสินใจและการทำงานของบุคคลโดยทั่วไป โดยที่คอมพายเลอร์แต่ละโปรแกรมก็จะทำหน้าที่ในการแปลภาษาของตนเอง (เช่น FORTRAN compiler ก็จะใช้แปลภาษาฟอร์แทรน เป็นต้น)

นอกจากนี้ยังได้พูดถึงขบวนการทำงานของคอมพายเลอร์ ตั้งแต่การอ่านโปรแกรมจนถึงการสร้างรหัสภาษาเครื่อง ภาษาที่ใช้ในการเขียนคอมพายเลอร์ รวมถึงความจำเป็นในการที่จะต้องมีคอมพายเลอร์ไว้ใช้ในงานคอมพิวเตอร์ โดยสรุปแล้ว คอมพายเลอร์เป็นเรื่องที่น่าทำการศึกษา ติดตาม และพัฒนา (ถ้าเป็นได้) อย่างยิ่งเรื่องหนึ่ง และเป็นแนวทางที่เป็นไปได้ในการที่จะสนับสนุนให้มีการขยายตัวในการนำเครื่องคอมพิวเตอร์มาใช้ในบ้านเรามากขึ้น (เช่นการมีคอมพายเลอร์ภาษาไทยที่มีคุณภาพดีและประสิทธิภาพสูง) ซึ่งจะเป็นประโยชน์ต่อการพัฒนาประเทศในส่วนของเครื่องมือที่จะนำมาใช้ เพื่อช่วยในด้านงานอาชีพของแต่ละบุคคลหรือกลุ่มบุคคลต่างๆ ต่อไป

เอกสารอ้างอิง

1. F.L. Bauer and J. Eickel, Editors (1976). "Compiler Construction," Springer - Verlag, Germany.
2. Alfred V. Aho and Jeffrey D. Ullman (1978). "Principles of Compiler Design, "Addison - Wesley Publishing Company, Inc. USA.
3. Philip M. Lewis II, Daniel J. Rosenkrantz and Richard E. Stearns (1978). "Compiler Design Theory," Addison - Wesley Publishing Company, Inc. USA.